Note to Self: Training IFS Mappings Repeatedly

This is only a note to self / speculative research idea, not a finished method or a claim that the approach works.

Core idea

- Consider an IFS-like collection of learnable linear or affine mappings:

$$M_1, M_2, \ldots, M_K.$$

- Instead of training one large composed system by ordinary end-to-end backpropagation, train whichever mapping happens to be selected using a local associative objective.

- For a training pair (x, y):

$$M_i x \approx y$$

for a randomly selected M_i.

- A simple local update could be ordinary gradient descent on

$$\|y - M_i x\|^2$$

or the corresponding outer-product update for a linear associative memory.

Repeated / iterated training

- After applying one mapping, use its output to create the input for another training step.

- Basic version:

$$x_0 \xrightarrow{M_i} x_1$$

then train another randomly selected mapping on

$$M_j x_1 \approx y.$$

- Continue repeatedly:

$$x_{t+1} = M_{i_t} x_t.$$

- The target y remains fixed while the current state x_t changes.

- This means later mappings learn to correct states produced by earlier mappings rather than always seeing the original input.

Possible random-intermediate version

- Instead of simply setting the next input to the target, insert a fixed or randomly selected transformation R:

$$x_{t+1} = RM_{i_t} x_t.$$

- Then train the next randomly selected mapping using

$$M_{i_{t+1}} x_{t+1} \approx y.$$

- A sequence could therefore look like

$$x \to M_i x \to RM_i x \to M_j RM_i x \to RM_j RM_i x \to \cdots$$

- This may force different mappings to learn complementary corrections rather than all simply learning the same direct mapping.

Important distinction

- Setting

$$x \leftarrow y$$

after every mapping is quite different.

- The next mapping then sees y itself and is effectively encouraged toward

$$M_j y \approx y,$$

i.e. an identity-like attractor around the targets.

- The random-R version seems potentially more interesting because the next mapping must recover from a transformed state.

Why this might connect to IFS

- An ordinary IFS repeatedly selects maps from a finite collection:

$$x_{t+1} = f_{i_t}(x_t).$$

- Here the maps would be learned rather than fixed:

$$x_{t+1} = M_{i_t} x_t.$$

- The difference is that the mappings are trained to **recover desired targets**, rather than merely being chosen to construct a geometric attractor.

- Possible reversal of the usual IFS viewpoint:

  Instead of mappings defining an attractor, train mappings so that their repeated random composition creates an attractor around desired states.

Questions to test

- Do the learned mappings become contractive?
- Do they develop common invariant subspaces?
- Do their products become progressively more stable?
- Does repeated random selection converge toward y?
- Do different mappings specialize in correcting different kinds of error?
- Does inserting R produce more interesting specialization than simply resetting to y?
- Can shears or other structured transformations be used for R?
- What happens when the mappings are affine,

$$M_i x + b_i,$$

rather than purely linear?

- Is there a useful balance between:

$$\text{individual mapping capacity} \quad \text{and} \quad \text{number of iterations?}$$

Possible connection to associative memory

- Each M_i is essentially a linear associative memory.
- Repeated training therefore amounts to repeatedly training small associative memories on states generated by other associative memories.
- This could provide a very simple experimental setting for studying how a population of local mappings organizes itself under repeated composition.

Note to Self: What I Noted About IFS as a Possible CCSLM Connection

Again, this is only a note to self summarizing a possible connection, not a literature claim that CCSLM is an IFS or that the two are equivalent.

- IFS is probably one of the closest established mathematical frameworks to the basic idea of a finite repertoire of simple transformations combined through repeated selection.
- An IFS has the form

$$x_{t+1} = f_{C_t}(x_t),$$

where a selection variable C_t chooses one transformation.

- CCSLM has the analogous structure of context selecting mappings:

$$x \mapsto L(C, x).$$

- **Self-affine IFSs** are especially relevant because their maps can contain general affine transformations rather than just simple similarities.

- Shears are entirely compatible with affine IFSs. A shear can be combined with contraction, for example,

$$A = s \begin{pmatrix} 1 & k\,0 & 1 \end{pmatrix}.$$

- Thus the mathematical machinery already exists for studying systems in which different selected transformations alter not only scale and position but also **coupling between dimensions**.

Possible CCSLM lessons from IFS

- Repeated selection of mappings can create global structure much richer than any individual mapping.

- The relevant object may therefore be

$$\text{mapping repertoire} + \text{selection dynamics} + \text{composition}.$$

- IFS shows that an attractor does not have to be a single point. It can be an invariant set produced by repeated compositions.

- This suggests that context-driven computational systems might also develop **stable regions, cycles, or attractor-like computational states**, particularly when context has temporal persistence.

- If context itself evolves according to the current state,

$$C_{t+1} = g(C_t, x_t),$$

then the combined context/data system becomes a dynamical system and may have its own attractors.

- There is an important warning: classical IFS attractor results often rely on contraction. Strong contraction may produce stable states while simultaneously destroying information.

- Therefore the CCSLM questions of **observability, rank, information preservation, and attractor formation** may be closely related but competing properties.

Tentative takeaway

- IFS suggests taking the **selection process itself** seriously rather than viewing each mapping in isolation.

- The most interesting CCSLM connection may not be "CCSLM is an IFS."
- It may instead be:

  An IFS demonstrates how a system consisting of simple transformations plus a selection mechanism can acquire rich global structure through composition.

- The speculative training idea above reverses the usual IFS direction:

  Train the individual transformations so that iterated selection and composition acquire useful attractor behavior.

- This seems simple enough to test directly with small matrices before drawing any theoretical conclusions.